



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/690,980	10/21/2003	Kenneth P. Sundermeyer	47583/P041US/10311286	4705

59061 7590 07/30/2007
FULBRIGHT & JAWORSKI, LLP (ADOBE)
2200 ROSS AVENUE
SUITE 2800
DALLAS, TX 75201-2784

EXAMINER

RUTLEDGE, AMELIA L

ART UNIT	PAPER NUMBER
----------	--------------

2176

MAIL DATE	DELIVERY MODE
-----------	---------------

07/30/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/690,980
Filing Date: October 21, 2003
Appellant(s): SUNDERMEYER ET AL.

MAILED

JUL 30 2007

Technology Center 2100

Jody C. Bishop
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed April 13, 2007 appealing from the Office action mailed 12/15/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

A substantially correct copy of appealed claim 15 appears on page 29 of the Appendix to the appellant's brief. The minor errors are as follows: lines 6-11 have not been updated to reflect the claim amendment filed 10/04/2006, and should contain the following limitations:

“restricting edit functions of a Web editor editing said substitute Web document, wherein said restricting is responsive to said one or more restriction markup tags; and displaying said non-editable content to provide context for an editable portion of said Web document.”

(8) Evidence Relied Upon

2004/0177321	BROWN et al.	9-2004
6,061,697	NAKAO	5-2000
5,752,643	Judson	11-1996

Dreamweaver TechNote 16416, "How to make an inherited editable region uneditable" July 6, 2002, pp. 1-4.

Macromedia Dreamweaver MX: Training from the Source, published July 23, 2002, Safari Tech Books Online version, <http://proquest.safaribooksonline.com>, p. 1-39.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brown et al. (hereinafter "Brown"), U.S. Pub. No. 2004/0177321, published September 2004, in view of *Dreamweaver TechNote16416*, "How to make an inherited editable region uneditable," (hereinafter "Dreamweaver"), last updated 07/06/2002, p. 1-4.

Regarding independent claim 1, Brown teaches a method of inserting XML restriction tags into an editable, i.e., revised, document (p. 9, par. 107-p. 10, par. 109) including instance level restrictions. Brown teaches a method of creating a bounding file or DTD to restrict editing of a formatted structured document (p. 8, par. 94-p. 9, par. 105), by parsing the input document and generating a bounding file (p. 6, par. 74-78). Therefore Brown teaches parsing the multi-formatted document, identifying a plurality of tags responsive to the parsing, and generating a revised document based on the multi-formatted document, and inserting restriction tags into the revised document delimiting non-editable content defined by one or more of the tags, because Brown teaches parsing an input document and generating a bounding document, a revised document

which can be edited and contains restriction tags delimiting non-editable content defined by the tags.

While Brown teaches that the restriction tags in the bounding document are used to determine non-editable content in either structured documents or a GUI, (p. 5, par. 63), i.e., restrict the functions of a page editor; Brown does not explicitly teach a method of restricting edit functions of a page editor editing the revised document where the restricting is responsive to the page editor reading the restriction tags. In other words, although Brown does teach generating an editable bounding file containing restriction tags, the bounding file *itself* is not edited by the page editor reading the restriction tags. Rather, the bounding file disclosed by Brown contains restriction tags to be used by the page editor for editing another document or GUI page.

However, Dreamweaver teaches the use of nested templates with editable and non-editable regions delimited by tags which are inserted into a revised document (p. 1-3), and which restrict the edit functions of the Dreamweaver page editor editing the revised document where the restricting is responsive to the page editor reading the restriction tags. In other words, the page editor disclosed by Dreamweaver reads the restriction tags which have been inserted into the revised document, in order to edit the revised document *itself* and determine non-editable content in the revised document.

The reference *Macromedia Dreamweaver MX: Training from the Source*, published July 23, 2002, Safari Tech Books Online version, <http://proquest.safaribooksonline.com>, p. 1-39; while not being relied upon for the

rejections, is cited for the purpose of providing additional information on Dreamweaver and Dreamweaver templates.

Both Brown and Dreamweaver are directed toward editing structured document files. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the templates with editable regions taught by Dreamweaver with the XML bounding DTD with restriction tags taught by Brown, so that the editing GUI would have functionality to control and restrict editing both of HTML (Dreamweaver) and XML (Brown), and so that the user would have the benefit of a simpler way to edit XML files and the application developer would have control over user actions while editing the files (Brown, p. 2, par. 18).

Regarding dependent claim 2, Brown teaches compiling a list of document locations containing editable content defined by the tags (Fig. 14).

Regarding dependent claim 3, Brown teaches hiding of restriction tags so that the user of a page editor cannot see the tags during editing (p. 6, par. 72).

Regarding dependent claims 4 and 5, Brown teaches setting a restriction flag in the revised document to activate edit restriction, by designating a tag editable or non-editable, which is read by the application (p. 8, par. 91-93).

Regarding dependent claim 6, Brown teaches the use of shading to notify the user of a restrictive editing location (p. 7, par. 80).

Regarding dependent claim 7, while Brown does not teach stripping out the restriction tags when said page editor writes said edited multi-formatted document for presentation, since brown teaches a bounding file, Dreamweaver teaches stripping out

Art Unit: 2176

the restriction tags when the document is written for presentation, since Dreamweaver teaches stripping out code from nested template documents (p. 3, "Making the region uneditable, but controlling the visibility of its content while in the nested template), and discloses that by modifying or moving code within an editable region, developers can control whether pieces of code are visible or hidden. Both Brown and Dreamweaver are directed toward editing structured document files. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the templates with editable regions taught by Dreamweaver with the XML bounding DTD with restriction tags taught by Brown, so that the editing GUI would have functionality to control and restrict editing both of HTML (Dreamweaver) and XML (Brown), and so that the user would have the benefit of a simpler way to edit XML files and the application developer would have control over user actions while editing the files (Brown, p. 2, par. 18).

In regard to independent claim 8, Brown teaches a method of inserting XML restriction tags into an editable document (p. 9, par. 107-p. 10, par. 109) including instance level restrictions. Brown teaches a method of creating a bounding file or DTD to restrict editing of a formatted structured document (p. 8, par. 94-p. 9, par. 105), by parsing the input document and generating a bounding file (p. 6, par. 74-78). Therefore Brown teaches parsing the multi-formatted document, identifying a plurality of labels, analyzing the labels, and generating a revised document using content of the multi-formatted document, because Brown teaches parsing an input document and generating a bounding document, a revised document which can be edited and contains restriction tags delimiting non-editable content defined by the tags.

While Brown teaches that the restriction tags in the bounding document are used to determine non-editable content in either structured documents or a GUI, (p. 5, par. 63), i.e., restrict the functions of a page editor; Brown does not explicitly teach code for inserting a prohibition label in said modified document around each instance of non-editable content as defined by said one or more descriptive labels; and code for prohibiting edit functions of a document editor editing said modified document, wherein said code for prohibiting is executed responsive to said prohibition labels. In other words, although Brown does teach generating an editable bounding file containing restriction tags, the bounding file *itself* is not edited by the page editor reading the restriction tags. Rather, the bounding file disclosed by Brown contains restriction tags to be used by the page editor for editing another document or GUI page.

However, Dreamweaver teaches the use of nested templates with editable and non-editable regions delimited by labels which are inserted into a revised document (p. 1-3), and which restrict the edit functions of the Dreamweaver page editor editing the revised document where the restricting is responsive to the page editor reading the restriction tags. In other words, the page editor disclosed by Dreamweaver reads the restriction tags which have been inserted into the revised document, in order to edit the revised document *itself* and determine non-editable content in the revised document. Therefore Dreamweaver teaches code for inserting a prohibition label in said modified document around each instance of non-editable content as defined by said one or more descriptive labels; and code for prohibiting edit functions of a document editor editing

said modified document, wherein said code for prohibiting is executed responsive to said prohibition labels.

The reference *Macromedia Dreamweaver MX: Training from the Source*, published July 23, 2002, Safari Tech Books Online version, <http://proquest.safaribooksonline.com>, p. 1-39; while not being relied upon for the rejections, is cited for the purpose of providing additional information on Dreamweaver and Dreamweaver templates.

Both Brown and Dreamweaver are directed toward editing structured document files. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the templates with editable regions taught by Dreamweaver with the XML bounding DTD with restriction tags taught by Brown, so that the editing GUI would have functionality to control and restrict editing both of HTML (Dreamweaver) and XML (Brown), and so that the user would have the benefit of a simpler way to edit XML files and the application developer would have control over user actions while editing the files (Brown, p. 2, par. 18).

Regarding dependent claims 9-14, claims 9-14 reflect substantially similar subject matter as claimed in dependent claims 2-7, being directed to the computer program product used to implement the methods as claimed in dependent claims 2-7, and are rejected along the same rationale.

Independent claim 15 cites: *A method to restrict editing of a Web document comprising: parsing said Web document; analyzing a plurality of markup tags within said Web document; generating a substitute Web document with content of said Web*

document; inserting one or more restriction markup tags in said substitute Web document demarcating non-editable content items defined by one or more of said plurality of markup tags; and restricting edit functions of a Web editor editing said substitute Web document, wherein said restricting is responsive to said one or more restriction markup tags; and displaying said non-editable content to provide context for an editable portion of said web document.

Brown teaches a method of inserting XML restriction tags into an editable, i.e., revised, document (p. 9, par. 107-p. 10, par. 109) including instance level restrictions. Brown teaches a method of creating a bounding file or DTD to restrict editing of a formatted structured document (p. 8, par. 94-p. 9, par. 105), by parsing the input document and generating a bounding file (p. 6, par. 74-78). Therefore Brown teaches parsing the multi-formatted document, identifying a plurality of tags within the document, and generating a revised document based on the multi-formatted document, and inserting restriction tags into the revised document delimiting non-editable content defined by one or more of the tags, because Brown teaches parsing an input document and generating a bounding document, a revised document which can be edited and contains restriction tags delimiting non-editable content defined by the tags.

While Brown teaches that the restriction tags in the bounding document are used to determine non-editable content in either structured documents or a GUI, (p. 5, par. 63), i.e., restrict the functions of a page editor; Brown does not explicitly teach a method of restricting edit functions of a Web editor editing said substitute Web document. In other words, although Brown does teach generating an editable bounding file containing

restriction tags, the bounding file *itself* is not edited by the page editor reading the restriction tags. Rather, the bounding file disclosed by Brown contains restriction tags to be used by the page editor for editing another document or GUI page.

However, Dreamweaver teaches the use of nested templates with editable and non-editable regions delimited by tags which are inserted into a revised document (p. 1-3), and which restrict the edit functions of the Dreamweaver page editor editing the revised document where the restricting is responsive to the page editor reading the restriction tags. In other words, the page editor disclosed by Dreamweaver reads the restriction tags which have been inserted into the revised document, in order to edit the revised document *itself* and determine non-editable content in the revised document.

Further, while Brown does not explicitly teach displaying said non-editable content to provide context for an editable portion of said web document, Dreamweaver discloses displaying the non-editable content to provide context for an editable portion of said web document, since Dreamweaver clearly discloses that both editable and non-editable content may be made visible to the user, as disclosed on p. 2-3. Dreamweaver explicitly teaches a method titled "Making the region uneditable, and retaining the visibility of its content while in the nested template", on p. 2.

The reference *Macromedia Dreamweaver MX: Training from the Source*, published July 23, 2002, Safari Tech Books Online version, <http://proquest.safaribooksonline.com>, p. 1-39; while not being relied upon for the rejections, is cited for the purpose of providing additional information on Dreamweaver and Dreamweaver templates.

Both Brown and Dreamweaver are directed toward editing structured document files. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the templates with editable regions taught by Dreamweaver with the XML bounding DTD with restriction tags taught by Brown, so that the editing GUI would have functionality to control and restrict editing both of HTML (Dreamweaver) and XML (Brown), and so that the user would have the benefit of a simpler way to edit XML files and the application developer would have control over user actions while editing the files (Brown, p. 2, par. 18).

Regarding dependent claims 16-21, claims 16-21 reflect substantially similar subject matter as claimed in dependent claims 2-7, and are rejected along the same rationale.

Claims 22-24, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nakao, U.S. Patent No. 6,061,697, issued May 2000, in view of Brown

Independent claim 22 cites: *A system for preserving design elements of a Web page during content editing, said system comprising: a Web development environment comprising: a parsing engine for analyzing a plurality of Web page markup tags; a list of restriction tags for insertion around said design elements, as defined by one or more of said plurality of Web page markup tags; and a page editor comprising: a plurality of deselectable editing functions, wherein said deselection is responsive to said restriction tags.*

Nakao teaches a SGML document editing apparatus which is implemented with a method of parsing a formatted SGML document (Col. 5, l. 5-22; Col. 7, l. 13-25) and identifying a plurality of editable tags in the document (Col. 9, l. 1-45). Nakao teaches a list of restriction tags for insertion around formatting elements (Col. 9, l. 1-45). While Nakao teaches a page editor, Nakao does not explicitly teach a page editor comprising a plurality of deselectable editing functions, where deselection is responsive to restriction tags. However, Brown teaches a web development environment and page editor with deselectable editing functions responsive to the restriction tags (p. 6, par. 72; p. 7, par. 80). Brown teaches a method of inserting XML restriction tags into the editable, i.e., revised, document (p. 9, par. 107-p. 10, par. 109) including instance level restrictions. Brown teaches a method of creating a bounding file or DTD to restrict editing of a formatted structured document (p. 8, par. 94-p. 9, par. 105), by parsing the input document and generating a bounding file (p. 6, par. 74-78).

Both inventions are directed toward restricting editing operations on structured documents. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Brown to Nakao, so that the user would have the benefit of techniques that enable application developers to have some control over what users can do when editing a structured document file (Brown p. 2, par. 21).

Regarding dependent claim 23, Nakao teaches a method of parsing a formatted SGML document (Col. 5, l. 5-22; Col. 7, l. 13-25) and identifying a plurality of editable tags in the document (Col. 9, l. 1-45); compare to: *a Web page is processed by said Web development environment to obtain said restriction tags.*

Regarding dependent claim 24, while Nakao does not explicitly teach that a subject matter expert operates said page editor to perform said content editing, Brown teaches that the editing system is designed for both application developers and users, where the users operate the restrictive page editor (p. 2, par. 21). The users taught by brown correspond to subject matter experts and would include that group. Both inventions are directed toward restricting editing operations on structured documents. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Brown to Nakao, so that the user would have the benefit of techniques that enable application developers to have some control over what users can do when editing a structured document file (Brown p. 2, par. 21).

Regarding dependent claim 26, while Nakao does not explicitly teach a restriction switch, Brown teaches a restriction switch for deselecting editing functionality (Figure 14) compare to *a restriction switch, accessible by said Web development environment, for activating deselectability of said plurality of deselectable editing functions*.

Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over Nakao in view of Brown, and further in view of Judson, U.S. Patent No. 5,752,643, issued November 1996.

Regarding dependent claim 25, while Nakao in view of Brown does not explicitly teach a cover object for obscuring the restriction tags from view, Judson teaches hiding tags in HTML comment tags and/or by the creation of a covering tag to

obscure another tag from view (Col. 5, l. 16-40), compare to *a cover object for obscuring said restriction tags from view in said page editor*. Nakao, Brown, and Judson are all directed toward the presentation and display of structured web documents. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Nakao, Brown, and Judson, so that the user would have the benefit of viewing the hidden information instantly, without the added time that would have been required to download the information, and information could be viewed as required by the user (Judson, col. 2, l. 12-28).

(10) Response to Argument

Beginning on page 9 of the Appeal Brief (hereinafter "the Brief"), Appellant argues the following issues which are accordingly addressed below.

A. Rejections Under 35 U.S.C. 103(a) over Brown in view of Dreamweaver.

1. "Lack of All Claimed Elements." (pages 9-20 of the Brief).

a. "Independent Claim 1 and Dependent Claims 2, 3, 5, and 6."

(pages 9-11 of the Brief).

Appellant argues that the combination of Brown and Dreamweaver does not teach the limitation of independent claim 1, "inserting one or more restriction tags into

said revised document delimiting non-editable content defined by one or more of said plurality of tags..." (Remarks, p. 9-11).

It is the examiner's opinion that both Brown and Dreamweaver teach the limitation, "inserting one or more restriction tags into said revised document delimiting non-editable content defined by one or more of said plurality of tags...", since both teach generating a new document containing restriction tags delimiting non-editable content, which suggests the combination of Brown and Dreamweaver. Dreamweaver is then relied upon to teach the limitation of claim 1, "restricting edit functions of a page editor editing said revised document, wherein said restricting is responsive to said page editor reading said one or more restriction tags," which was not disclosed by Brown.

As set forth in the rejection of claim 1 above, Brown teaches a method of creating a bounding file or DTD to restrict editing of a formatted structured document (p. 8, par. 94-p. 9, par. 105), by parsing the input document and generating a bounding file (p. 6, par. 74-78). Therefore Brown teaches parsing the multi-formatted document, identifying a plurality of tags responsive to the parsing, and generating a revised document based on the multi-formatted document, and inserting restriction tags into the revised document delimiting non-editable content defined by one or more of the tags, because Brown teaches parsing an input document and generating a new, revised, bounding file containing restriction tags.

Dreamweaver teaches the use of nested templates with editable and non-editable regions delimited by tags which are inserted into a revised document (p. 1-3), and which restrict the edit functions of the Dreamweaver page editor editing the revised

document where the restricting is responsive to the page editor reading the restriction tags. Dreamweaver teaches using a template to generate a revised template document containing restriction tags designating non-editable regions, which are then read by the page editor (p. 1, par. 4-5).

While Appellants argue that Dreamweaver's editing restrictions are inserted into a separate nested template file, and not into an instance of the template, i.e., a revised document (p. 10, par. 2-3 of the Brief), Appellants assertion is clearly contracted by Dreamweaver, p. 1, par. 4-5, which recites, in part, "In the figure below, the editable regions called EditRegion1 and EditRegion2 are from the main template. EditRegion3 was added in the nested template. Since EditRegion3 was added in the nested template to an editable region (EditRegion2) already available from the main template, EditRegion2 will not be editable in the instances of the nested template." Therefore Dreamweaver explicitly teaches inserting editing restrictions, designated by restriction tags, into an instance of a template, which is a revised document.

It is also respectfully noted that appellants provided no explicit definition for the term "revised document" in the original disclosure. Where an explicit definition is provided by the appellant for a term, that definition will control interpretation of the term as it is used in the claim. *Toro Co. v. WhiteConsolidated Industries Inc.*, 199 F.3d 1295, 1301, 53 USPQ2d 1065, 1069 (Fed. Cir. 1999). Since no definition was provided the claim term must be given the broadest reasonable interpretation in light of the supporting disclosure, in this case the term "revised document" is interpreted to mean a new or updated document, template, or document file.

b. “Dependent Claim 4” (p. 11-12 of the Brief).

Appellant argues that dependent claim 4 does not teach the limitation “setting a restriction flag in said revised document to activate edit restriction.” Appellants arguments are based on the assertion that Brown does not teach inserting tags in a revised document in the limitations of claim 1 from which claim 4 depends. However, Brown does teach inserting restriction tags in a revised document as set forth in the response to arguments for claim 1.

Brown teaches a method of creating a bounding file or DTD to restrict editing of a formatted structured document (p. 8, par. 94-p. 9, par. 105), by parsing the input document and generating a bounding file (p. 6, par. 74-78). Therefore Brown teaches parsing the multi-formatted document, identifying a plurality of tags responsive to the parsing, and generating a revised document based on the multi-formatted document, and inserting restriction tags into the revised document delimiting non-editable content defined by one or more of the tags, because Brown teaches parsing an input document and generating a new, revised, bounding file containing restriction tags.

c. “Dependent Claim 7” (p. 12-13 of the Brief).

In response to Appellant’s arguments that Dreamweaver does not teach “stripping out one or more restriction tags when said page editor writes said edited

multi-formatted document for presentation,” Dreamweaver does teach stripping out the restriction tags when the document is written for presentation, since Dreamweaver teaches stripping out code from nested template documents (p. 3, “Making the region uneditable, but controlling the visibility of its content while in the nested template), and discloses that by modifying or moving code within an editable region, developers can control whether pieces of code are visible or hidden in the template instance when the document is presented, thereby stripping out the restriction tags.

d. “Independent Claim 8 and Dependent Claims 9, 10, 12, and 13” (p. 13-15 of the Brief).

Appellant argues that the combination of Brown and Dreamweaver does not teach the limitation of independent claim 1, “code for inserting a prohibition label in said modified document around each instance of non-editable content as defined by said one or more descriptive labels...” (Remarks, p. 13-15).

It is the examiner’s opinion that both Brown and Dreamweaver teach the limitation, “code for inserting a prohibition label in said modified document around each instance of non-editable content as defined by said one or more descriptive labels...”, since both teach generating a new document containing restriction tags, or labels, delimiting non-editable content, which suggests the combination of Brown and Dreamweaver. Dreamweaver is then relied upon to teach the limitation of claim , “code for prohibiting edit functions of a document editor editing said modified document,

wherein said code for prohibiting is executed responsive to said prohibition labels," which was not disclosed by Brown.

As set forth in the rejection of claim 8 above, Brown teaches a method of creating a bounding file or DTD to restrict editing of a formatted structured document (p. 8, par. 94-p. 9, par. 105), by parsing the input document and generating a bounding file (p. 6, par. 74-78). Therefore Brown teaches parsing the multi-formatted document, identifying a plurality of tags responsive to the parsing, and generating a revised document based on the multi-formatted document, and inserting restriction tags into the revised document delimiting non-editable content defined by one or more of the tags, because Brown teaches parsing an input document and generating a new, revised, bounding file containing restriction tags.

Dreamweaver teaches the use of nested templates with editable and non-editable regions delimited by labels, i.e., tags which are inserted into a revised document (p. 1-3), and which restrict the edit functions of the Dreamweaver page editor editing the revised document where the restricting is responsive to the page editor reading the restriction tags. Dreamweaver teaches using a template to generate a revised template document containing restriction tags designating non-editable regions, which are then read by the page editor (p. 1, par. 4-5).

While Appellants argue that Dreamweaver's editing restrictions are inserted into a separate nested template file, and not into an instance of the template, i.e., a revised document (p. 14, par. 1 of the Brief), Appellants assertion is clearly contracted by Dreamweaver, p. 1, par. 4-5, which recites, in part, "In the figure below, the editable

regions called EditRegion1 and EditRegion2 are from the main template. EditRegion3 was added in the nested template. Since EditRegion3 was added in the nested template to an editable region (EditRegion2) already available from the main template, EditRegion2 will not be editable in the instances of the nested template.” Therefore Dreamweaver explicitly teaches inserting editing restrictions, designated by restriction tags, into an instance of a template, which is a revised document.

It is also respectfully noted that appellants provided no explicit definition for the term “revised document” in the original disclosure. Where an explicit definition is provided by the appellant for a term, that definition will control interpretation of the term as it is used in the claim. *Toro Co. v. WhiteConsolidated Industries Inc.*, 199 F.3d 1295, 1301, 53 USPQ2d 1065, 1069 (Fed. Cir. 1999). Since no definition was provided the claim term must be given the broadest reasonable interpretation in light of the supporting disclosure, in this case the term “revised document” is interpreted to mean a new or updated document, template, or document file.

e. “Dependent Claim 11” (p. 15 of the Brief).

Appellant argues that dependent claim 11 does not teach the limitation “code for setting a restriction feature flag in said modified document to activate restricted editing.” Appellants arguments are based on the assertion that Brown does not teach inserting labels in a revised document in the limitations of claim 1 from which claim 4 depends.

However, Brown does teach inserting restriction tags in a revised document as set forth in the response to arguments for claim 1.

Brown teaches a method of creating a bounding file or DTD to restrict editing of a formatted structured document (p. 8, par. 94-p. 9, par. 105), by parsing the input document and generating a bounding file (p. 6, par. 74-78). Therefore Brown teaches parsing the multi-formatted document, identifying a plurality of tags responsive to the parsing, and generating a revised document based on the multi-formatted document, and inserting restriction tags into the revised document delimiting non-editable content defined by one or more of the tags, because Brown teaches parsing an input document and generating a new, revised, bounding file containing restriction tags.

f. “Dependent Claim 14” (p. 16 of the Brief).

In response to Appellant’s arguments that Dreamweaver does not teach “code for removing said prohibition tags prior to saving said modified multi-formatted document for presentation,” Dreamweaver does teach removing said prohibition tags prior to saving said modified multi-formatted document for presentation, since Dreamweaver teaches stripping out code from nested template documents (p. 3, “Making the region uneditable, but controlling the visibility of its content while in the nested template), and discloses that by modifying or moving code within an editable region, developers can control whether pieces of code are visible or hidden in the

template instance when the document is presented, thereby removing the prohibition tags.

- g. “Independent Claim 15 and Dependent Claims 16, 17, 19, and 20” (p. 16-18 of the Brief).**
- h. “Dependent Claim 18” (p. 18-19 of the Brief).**
- i. “Dependent Claim 21” (p. 19-20 of the Brief).**

Appellant’s arguments for independent claim 15 and dependent claims 16-21 follow a substantially similar rationale to Appellant’s arguments presented for claims 1-7 and 9-14, respectively, and for similar reasons as set forth in the responses to arguments for claims 1-7 and 9-14, above, the rejections should be maintained.

2. “Lack of Motivation.” (pages 20-21 of the Brief).

In response to Appellant’s argument in regard to the motivation statement for the combination of Brown and Dreamweaver (page 20 of the Brief), Appellant’s arguments question the validity of the motivation to combine the references because the motivation statement cited Brown, the primary reference. However, there is no requirement for the motivation to combine the references to come from any particular reference, since obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or

motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case, the suggestion may be found in Brown, p. 2, par. 18. The cited passage of Brown reads: *Accordingly, what is needed is a cost-effective way to enable users to edit XML files (or files encoded in other markup languages), while shielding them from details of the XML language (e.g., the tags and attributes of a particular data model) and at the same time, enabling application developers to have some control over what the users can do when they are editing XML files.* Therefore, contrary to appellants' argument, Brown does indicate a need to control and restrict editing both of HTML and XML, i.e., *what is needed is a cost-effective way to enable users to edit XML files (or files encoded in other markup languages)* and therefore does provide motivation for the combination of references.

Both Brown and Dreamweaver are directed toward editing structured document files. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the templates with editable regions taught by Dreamweaver with the XML bounding DTD with restriction tags taught by Brown, so that the editing GUI would have functionality to control and restrict editing both of HTML (Dreamweaver) and XML (Brown), and so that the user would have the benefit of a simpler way to edit XML files and the application developer would have control over user actions while editing the files (Brown, p. 2, par. 18).

B. Rejections Under 35 U.S.C. 103(a) over Nakao in view of Brown.

1. “Lack of All Claimed Elements.” (pages 21-23 of the Brief).

In response to Appellant's, arguments regarding claim 22-24 and 26, and in regard to the Nakao reference (Remarks, p. 21-23), the passages of Nakao relied upon for the basis of Appellants' arguments refer to the Background of Invention section of the reference (col. 1, l. 1-col. 4, l. 63), not the disclosure of Nakao. Therefore it appears that Appellants' arguments related to the cited passage are moot (p. 21-22 of the Brief). It is the examiner's opinion that the combination of Nakao and Brown discloses the limitation *identifying a plurality of web page markup tags*, for the reasons set forth in the rejection of claim 22, above, and in the citations relied upon for the rejection.

Appellant argues that if document elements are subject to being edited, the elements must not be equivalent to tags (p. 21-22 of the Brief). However, since tags could themselves be edited by being updated and modified, the logic of this argument presented by Appellant is flawed.

Further, even if the Duckett reference cited by Appellant were correct (page 22, par. 3-4 of the Brief), in stating “elements are tags and anything between the opening and closing tags”, the elements would still, by Appellant's definition, include tags.

Therefore Nakao does teach the limitation of claim 22, “a parsing engine for analyzing a plurality of Web page markup tags” at col. 5, l. 5-22 and col. 7, l. 13-25. Brown also teaches parsing tags in a document page, at p. 8, par. 0094.

In response to Appellant's arguments that Brown does not teach the limitation of claim 22, "a plurality of deselectable editing functions, wherein said deselection is responsive to said restriction tags," In response to Appellant's argument that the references fail to show certain features of Appellant's invention, it is noted that the features upon which Appellant relies (i.e., the reading of a restriction tag, page 23, par. 1, l. 6 of the brief) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

2. "Lack of Motivation." (pages 23-24 of the Brief).

In response to Appellant's argument in regard to the motivation statement for the combination of Nakao and Brown (Remarks, p. 14), obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case, the suggestion may be found in Brown, p. 2, par. 21.

Both inventions are directed toward restricting editing operations on structured documents. It would have been obvious to one of ordinary skill in the art at the time of the invention to apply Brown to Nakao, so that the user would have the benefit of

techniques that enable application developers to have some control over what users can do when editing a structured document file (Brown p. 2, par. 21).

C. Rejection Under 35 U.S.C. 103(a) over Nakao in view of Brown, and further in view of Judson.

1. "Lack of All Claimed Elements." (pages 24-25 of the Brief).

Appellant's arguments regarding the Judson patent relied upon to reject dependent claim 25 do not dispute the teaching of Judson but rather only refer back to the purported deficiencies of Nakao and Brown, which have been addressed above.

2. "Lack of Motivation." (pages 25-26 of the Brief).

In response to Appellant's arguments in regard to the motivation statement for the combination of Nakao, Brown, and Judson, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the suggestion may be found in Judson, col. 2, l. 12-28.

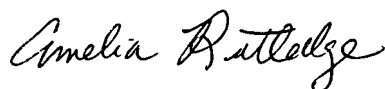
Nakao, Brown, and Judson are all directed toward the presentation and display of structured web documents. It would have been obvious to one of ordinary skill in the art at the time of the invention to combine Nakao, Brown, and Judson, so that the user would have the benefit of viewing the hidden information instantly, without the added time that would have been required to download the information, and information could be viewed as required by the user (Judson, col. 2, l. 12-28). While Appellants argue that the motivation for combining Nakao, Brown, and Judson is circular, it is the examiner's opinion that the desirability of instantly viewing hidden information in a markup document editor, as required by the user, would have made the combination of Judson with Nakao and Brown both obvious and desirable to one of ordinary skill in the art at the time of the invention.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

A handwritten signature in cursive script, reading "Amelia Rutledge".

Amelia Rutledge

July 23, 2007

Conferees:



Doug Hutton (Supervisor)



Stephen Heng (Supervisor)

William Bashore (Primary Examiner)



**WILLIAM BASHORE
PRIMARY EXAMINER**